

Claims

What is claimed is:

1. A system for interacting with computer programming languages comprising:
a plurality of objects that enable interactions with computer programming languages at a semantic level.
2. The system of claim 1, the plurality of objects further enabling interactions with computer programming languages at a syntactic level.
3. The system of claim 1, wherein the plurality of objects are associated with a code model to enable interactions with the computer programming languages.
4. The system of claim 3, wherein the code model further comprises a plurality of code model objects for encapsulating functionality within the computer programming languages.
5. The system of claim 4, wherein the code model models at least one of a syntax, one or more functions, one or more objects one or more data structures associated with the computer programming language.
6. The system of claim 4, wherein the plurality of code model objects are defined by a set of object classes to enable modeling and abstraction of the computer programming language.
7. The system of claim 4, wherein the code model objects model semantic elements within the computer programming language.
8. The system of claim 7, wherein the semantic elements include at least one of an attribute, a method, a type, an interface, a statement, an expression and a parameter.

10. The system of claim 9 further comprising a programmatic interface, wherein the programmatic interface includes methods for interacting with the set of object classes.

12. The system of claim 9, further comprising a CodeVariable object for modeling variables within a project.

20

14. A method for interacting with a computer programming language at a semantic level comprising :

retrieving a code model object; and
interacting with the code model object.

15. The method of claim 14 further comprising :

interacting with the code model object by reading one or more attributes of the code model object;

further interacting with the code model object by invoking one or more methods of the code model object;

further interacting with the code model object by adding and/or removing zero or more items from the code model object; and

further interacting with the code model object by incorporating the code model object into a computer programming project.

16. The method of claim 15 wherein the items comprise interfaces, base classes, namespaces, classes, functions, statements, expressions, variables, structures, enumerations, attributes, methods and delegates.

17. A system for interacting with one or more programming languages, comprising:

means for retrieving a code model object; and
means for interacting with the code model object.

18. The system of claim 17 wherein the means for interacting with one or more programming languages further comprises:

means for reading the one or more programming languages; and
means for writing to the one or more programming languages.

19. The system of claim 17 wherein the means for interacting with the code model object further comprises:

means for reading one or more attributes of the code model object;
means for writing one or more attributes of the code model object; and
means for invoking one or more methods of the code model object.

20. The system of claim 17 wherein the means for interacting with the code model object further comprises:

means for adding one or more items from the code model object; and
means for removing one or more items from the code model object.

21. The system of claim 20 wherein the items comprise interfaces, base classes, namespaces, classes, functions, variables, structures, enumerations, delegates, attributes and methods.

22. A system for interacting with computer programming languages at a semantic and syntactic level comprising:

a code model associated with a plurality of code model objects to enable a programmatic interface to interact with a plurality of computer programming languages at a semantic and syntactic level, wherein the code model provides isolation between the programmatic interface and the computer programming languages.

23. The system of claim 22, wherein the code model objects encapsulate functionality within the computer programming languages.

24. The system of claim 23, wherein the code model describes at least one of a syntax, one or more functions, one or more objects and one or more data structures associated with the computer programming languages.

25. The system of claim 23, wherein the code model objects are defined by a set of object classes to enable modeling and abstraction of the computer programming languages.

26. The system of claim 25, wherein the code model objects model semantic elements within the computer programming languages.

27. The system of claim 26, wherein the semantic elements include at least one of an attribute, a method, a type, an interface, and a parameter.

28. The system of claim 25, wherein the set of object classes further comprise at least one of a project item, a project, a code model, a file code model, a code element, a code name space, a code type, a code class a code interface, a code enumeration, a code delegate, a code function, a code parameter, a code attribute, a code statement, a code expression, a code structure, a code variable, a code property, a code type reference.

29. The system of claim 25, wherein set of object classes associated with the computer programming languages are modified to become the code model objects *via* at least one of an add code method, a delete code method, and coding *via* a text buffer.

30. The system of claim 25, further comprising a CodeVariable object for modeling variables within a project.

31. The system of claim 30, wherein the code variable object further comprises attributes that include at least one of an attribute for storing a parent of the CodeVariable object, an attribute for storing the initial expression and value for the variable modeled by the CodeVariable object, an attribute for storing the type of the variable modeled by the CodeVariable object, an attribute for storing whether the variable belongs to a class or to instances of a class, and an attribute for determining whether the variable is a constant.